

| Narration  | Time  |
|--|-------|
| Ruby میں Object Oriented Concept کے اسپوکن ٹیوٹوریل میں خوش آمدید۔                             | 00:01 |
| اس ٹیوٹوریل میں ہم   | 00:06 |
| classes استعمال کرنا   | 00:08 |
| objects بنانا  | 00:09 |
| Ruby میں methods واضح کرنے کے مختلف طریقے سیکھیں گے۔   | 00:10 |
| یہاں ہم  | 00:13 |
| Ubuntu ورژن 12.04 اور  | 00:14 |
| Ruby 1.9.3 استعمال کر رہے ہیں  | 00:16 |
| اس ٹیوٹوریل کو سمجھنے کے لئے آپ کو لینکس کمانڈس، ٹرمینل اور ٹیکسٹ ایڈیٹر کا علم ہونا ضروری ہے۔ | 00:19 |
| اگر نہیں، تو متعلقہ ٹیوٹوریل کے لئے، ہماری ویب سائٹ پر جائیں۔                                  | 00:24 |
| شروع کرنے سے پہلے، یاد کریں کہ ہم نے پہلے ہی ttt ڈائریکٹری بنائی تھی۔                          | 00:28 |
| اب اس ڈائریکٹری پر جاتے ہیں۔   | 00:33 |
| ruby hyphen tutorial اور classes ڈائریکٹری پر  | 00:35 |
| Ruby ایک object oriented لینگویج یعنی زبان ہے۔   | 00:41 |
| Ruby میں ویلیو سے string یا نمبر تک، سب کچھ ایک آبجیکٹ ہے؛                                     | 00:44 |
| ایک class، متعلقہ data اور functions کا مجموعہ ہے۔   | 00:49 |
| یہ منظم معلومات رکھنے کے کام آ سکتا ہے۔  | 00:53 |
| ایک object، class کی ایک instantiation ہے۔   | 00:56 |
| ایک class کی وضاحت کی وارڈ class سے شروع ہوتی ہے۔  | 01:00 |
| اس کے بعد class کا نام آتا ہے۔   | 01:05 |
| یہ end کے ساتھ ختم ہونے والی ہوتی ہے۔  | 01:08 |
| اب class کی ایک مثال دیکھتے ہیں۔   | 01:11 |
| class Product  | 01:14 |

|       |  |
|-------|--|
| 01:16 | ruby کوڈ   |
| 01:17 | end  |
| 01:20 | class نام کیپٹل لیٹر سے شروع ہونا چاہئے۔   |
| 01:24 | نام، جس میں ایک سے زیادہ الفاظ ہوں، camel case میں ہونے چاہئیں۔                  |
| 01:28 | مثلاً،   |
| 01:30 | UserInformation  |
| 01:32 | ProductInformation   |
| 01:34 | بعد والی فائلوں کے نام کے الفاظ کو الگ کرنے میں underscore کا استعمال ہوتا ہے۔   |
| 01:37 | user underscore information  |
| 01:40 | product underscore information   |
| 01:45 | Ruby کے ابتدائی ٹیوٹوریلز میں ظاہر طریقے سے ہی gedit میں ایک نئی فائل بنائیں۔    |
| 01:48 | اس کو class_definition.rb نام دیں۔   |
| 01:52 | میرے پاس classes کے نفاذ کی ایک فعال مثال ہے۔                                    |
| 01:57 | ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں               |
| 02:02 | میں نے اس مثال میں کلاس نامی Order کو ڈفائن یعنی واضح کیا ہے۔                    |
| 02:05 | اب، کچھ ویریبلس جوڑ کر class کو فائدہ مند بناتے ہیں۔                             |
| 02:11 | پھر میں نے ایک instance variable ، myinstance ڈفائن کیا ہے۔                      |
| 02:15 | اور اس کو ایک ویلیو مقرر کی ہے   |
| 02:18 | میں نے ایک کلاس وریبل myclassvar بھی ڈفائن کیا ہے۔                               |
| 02:21 | اور اس کو ایک ویلیو متعین کیا ہے۔  |
| 02:24 | اب، class کے استعمال کے لئے کچھ کوڈ شامل کرتے ہیں۔                               |
| 02:30 | ٹائپ کریں، puts Order dot instance underscore variables                          |
| 02:36 | اس لائن سے پہلے، ایک نئی لائن کے لئے puts، کچھ کیریکٹرز سلش n شامل کریں۔         |
| 02:43 | اس کو کاپی، پیسٹ کریں اور جوڑی ہوئی لائن کے نیچے اسے شامل کریں اور اسے سیو کریں۔ |

|       |   |
|-------|---|
| 02:51 | اب اس کو ڈکوائیز کیوٹ کریں۔   |
| 02:53 | ٹرمینل پر جائیں اور ٹائپ کریں   |
| 02:56 | ruby space class underscore definition dot rb                         |
| 03:02 | اور آؤٹ پٹ دیکھیں۔  |
| 03:05 | آپ کو اپنا واضح کیا ہوا instance variable نظر آئے گا۔                 |
| 03:09 | اب ٹائپ کریں puts Order dot class underscore variables.               |
| 03:15 | اس حد بندی کو، لائن کے بالکل نیچے کاپی پیسٹ کریں اور سیو کریں۔        |
| 03:21 | اب ٹرمینل پر جائیں اور پہلے کی طرح ہی فائل کو ایکز کیوٹ کریں۔         |
| 03:26 | آپ اپنا واضح کیا ہوا، class variable دیکھیں گے،                       |
| 03:32 | اب آپ اپنا class لکھنے کے قابل ہیں۔                                   |
| 03:35 | آگے، دیکھتے ہیں object کیا ہے؟  |
| 03:40 | ایک object کلاس کا اسٹیس ہے۔  |
| 03:43 | جس کا مطلب، object، کلاس سے بنتا ہے۔                                  |
| 03:46 | object، کلاس میں وضاحت شدہ خصوصیتیں اور میٹھڈس رکھے گا۔               |
| 03:52 | آپ object کس طرح ڈکلیئر کرتے ہیں                                      |
| 03:54 | ہم، ایک نیا کی وارڈ استعمال کرتے ہوئے کلاس کا object ڈکلیئر کرتے ہیں۔ |
| 03:58 | یہاں ہم Product کلاس کا object ڈکلیئر کر رہے ہیں۔                     |
| 04:02 | یہاں object بن گیا ہے۔  |
| 04:05 | product = Product.new   |
| 04:09 | یہ عمل object کی initialization کہلاتا ہے۔                            |
| 04:12 | یہ object، Product کی قسم ہے۔   |
| 04:16 | اب دیکھتے ہیں method "initialize" کیا ہے؟                             |
| 04:20 | initialize method، object بناتے وقت call ہوتا ہے۔                     |
| 04:26 | object پر new، کال کرنے سے ہم initialize method طلب کرتے ہیں۔         |

|       |  |
|-------|--|
| 04:31 | initialize method کے پاس parameters کی فہرست ہو سکتی ہے۔                               |
| 04:37 | دیگر Ruby میٹھڈس کی طرح، یہ def کی وارڈ کے بعد آتا ہے۔                                 |
| 04:43 | ایک مثال دیکھتے ہیں۔   |
| 04:46 | ابتدائی سطح کے Ruby ٹیوٹوریلز کی طرح ہی gedit میں ایک نئی فائل بنائیں۔                 |
| 04:50 | اور اس کو object underscore initialize dot rb نام دیں۔                                 |
| 04:55 | میرے پاس، object initialization کوڈ کی ایک فعال مثال ہے۔                               |
| 05:00 | ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں                     |
| 05:04 | یہاں میں نے ایک Order نامی کلاس کو ڈفائن کیا ہے۔                                       |
| 05:08 | پھر بغیر آرگومینٹ کے میں نے ایک initialize method کو واضح کیا ہے۔                      |
| 05:13 | میں نے ایک میسج I have created an object ظاہر کرنے کے لئے ایک puts میٹھڈ کو ڈفائن کیا، |
| 05:20 | مزید، میں نے Order dot new کو ڈفائن کیا۔   |
| 05:24 | یہ initialize method طلب کرے گا۔   |
| 05:27 | terminal پر جائیں اور ٹائپ کریں۔   |
| 05:31 | ruby space object underscore initialize dot rb   |
| 05:36 | اور آؤٹ پٹ دیکھیں۔   |
| 05:39 | آپ کو میسج نظر آئے گی۔ I have created an object  |
| 05:43 | اب واپس gedit پر جائیں اور میٹھڈ میں ایک argument شامل کریں۔                           |
| 05:48 | اب puts میں تبدیلی کرتے ہیں۔   |
| 05:51 | یہ پاس کئے ہوئے argument کی ویلیو ظاہر کرنا چاہئے۔                                     |
| 05:55 | آگے ٹائپ کریں  |
| 05:56 | Order dot new("I have created an object")۔   |
| 06:04 | یہاں ہم نے method "new" کو ایک argument دیا ہے۔  |
| 06:08 | یہ argument، initialize method کو پاس کیا جاتا ہے۔                                     |
| 06:13 | ٹرینل پر جائیں اور ٹائپ کریں   |

|   |       |
|---|-------|
| ruby space object underscore initialize dot rb                            | 06:16 |
| اور آؤٹ پٹ دیکھیں۔  | 06:20 |
| آپ پرنٹ کیا ہوا میسج، "I have created an object" دیکھیں گے۔               | 06:22 |
| اب، آپ object initialization کا مطلب سمجھ چکے ہوں گے۔                     | 06:29 |
| یاد کریں، Ruby میں methods، functions ہیں جو class ظاہر کرتے ہیں۔         | 06:33 |
| class میں ہر طریقہ def اور end بلاک میں ڈفائن کیا ہوا ہوتا ہے۔            | 06:39 |
| ملٹی ورڈ میٹھڈ نیم، underscore سے الگ کیا ہوا ہوتا ہے۔                    | 06:45 |
| کچھ characters جو Method نیم کے ساتھ منسلک کئے جاسکتے ہیں مندرجہ ذیل ہیں: | 06:48 |
| (question-mark)?  | 06:54 |
| = (equal to)  | 06:56 |
| ہر کریکٹر، method میں کچھ مطلب جوڑتا ہے۔                                  | 06:58 |
| اب کچھ مثالیں دیکھتے ہیں۔   | 07:02 |
| ابتدائی سطح کے Ruby ٹیوٹوریلز کی طرح ہی gedit میں ایک نئی فائل بنائیں۔    | 07:05 |
| اور اس کو class underscore methods dot rb نام دیں۔                        | 07:09 |
| میرے پاس کلاس میٹھڈس کوڈ کی ایک فعال مثال ہے۔                             | 07:14 |
| آپ ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں     | 07:17 |
| یہاں میں نے Animal نامی ایک class کو ڈفائن کیا۔                           | 07:21 |
| پھر میرے پاس دو میٹھڈس ہیں breathe اور walk                               | 07:23 |
| یہ دونوں def اور end کی وارڈس کے ساتھ ڈفائن ہوئے ہیں۔                     | 07:28 |
| پھر میں نے Animal object، انشلاز کیا۔                                     | 07:32 |
| میں نے اسے، ویریبل animal متعین کیا، چھوٹے a کے ساتھ۔                     | 07:36 |
| پھر میں نے breathe اور walk میٹھڈس کو ایک کے بعد ایک طلب کیا۔             | 07:40 |
| اب پروگرام کو ایکزیکوٹ کریں۔  | 07:48 |
| ٹرینل کھولیں اور ٹائپ کریں۔   | 07:51 |

|  |       |
|--|-------|
| ruby space class underscore methods dot rb   | 07:53 |
| اور آؤٹ پٹ دیکھیں  | 07:58 |
| آپ کو لائنز نظر آئیں گی:   | 08:00 |
| "I breathe"  | 08:02 |
| "I walk"   | 08:03 |
| پرنٹ ہوئی ہیں۔   | 08:04 |
| یہ اس لئے ہے کیونکہ آپ نے دو میٹھڈس breathe اور walk استعمال کئے ہیں۔                                      | 08:05 |
| ، ان میٹھڈس میں ڈفائن کیا ہوا puts اسٹیٹمنٹ نتیجہ دیتا ہے، جو آپ دیکھتے ہیں۔                               | 08:10 |
| مزید، دیکھتے ہیں کہ پشت پر question مارک کے ساتھ والے میٹھڈس کس طرح بناتے ہیں۔                             | 08:16 |
| ابتدائی سطح کے Ruby ٹیوٹوریل کی طرح ہی gedit میں ایک نئی فائل بنائیں۔                                      | 08:21 |
| اور اسے class underscore methods underscore with underscore trailing underscore characters dot rb نام دیں۔ | 08:25 |
| میرے پاس question مارک کوڈ کے ساتھ کلاس میٹھڈس کی ایک فعال مثال ہے۔  | 08:35 |
| ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں   | 08:40 |
| یہاں مثال کے طور پر میں نے پہلے والی ہی class لی ہے۔   | 08:45 |
| یہاں breathe میٹھڈ پشت پر question mark رکھتا ہے۔  | 08:48 |
| ایسے میٹھڈس عام boolean ویلیوز رٹرن کرنے میں استعمال ہوتے ہیں۔   | 08:52 |
| یہ Ruby کے ہینگ کنونشن میٹھڈ پر مبنی ہے۔   | 08:55 |
| animal dot breathe question-mark، ڈکلیئر کر کے میٹھڈ invoke یعنی طلب ہوتا ہے۔                              | 09:00 |
| ٹرینل کھولیں اور ٹائپ کریں   | 09:06 |
| ruby space class underscore methods underscore with underscore trailing underscore characters dot rb       | 09:09 |
| آپ کو آؤٹ پٹ میں True نظر آئے گا۔  | 09:22 |
| مزید، ہم دوسرے method، walk کو ڈفائن کرتے ہیں۔   | 09:26 |

|       |   |
|-------|---|
| 09:30 | اس کے پاس ایک "=(value) sign "equal-to رکھتے ہیں  |
| 09:36 | animal dot walk، کال کر کے اس کو استعمال کریں۔  |
| 09:41 | پھر اس میتھڈ کو ایکز کیوٹ کریں۔   |
| 09:44 | ٹرمینل کھولیں اور ٹائپ کریں   |
| 09:45 | ruby class underscore methods underscore with underscore trailing<br>underscore characters dot rb |
| 09:52 | اور آؤٹ پٹ دیکھیں۔  |
| 09:56 | یہ ایک "undefined method" ایرر دیگا   |
| 09:59 | یہ اس لئے ہے کیونکہ equal to سائن مختلف مطلب رکھتا ہے۔  |
| 10:03 | یہ ایک میتھڈ کو ویلیو مقرر کرنے میں استعمال ہوتا ہے۔  |
| 10:08 | اب، اس بار میتھڈ کو کچھ مختلف طریقے سے استعمال کرتے ہیں۔  |
| 10:13 | ٹائپ کریں "hops" equal to animal dot walk puts  |
| 10:17 | اب، ایک دوسری کوشش کریں۔  |
| 10:20 | ٹرمینل پر جائیں اور پہلے کی طرح ہی کمانڈ رن کریں اور آؤٹ پٹ دیکھیں۔                               |
| 10:27 | ہم دیکھیں گے کہ لفظ hops پرنٹ ہوا ہے۔   |
| 10:30 | یہ ظاہر کرتا ہے کہ میتھڈ کے آگے equal ٹوسائن کا مطلب ہے، تفویض۔                                   |
| 10:36 | اب آپ اپنے میتھڈس کو لکھنے کے قابل ہونا چاہیے۔  |
| 10:42 | اس ٹیوٹوریل میں ہم نے سیکھا۔  |
| 10:44 | classes کس طرح ڈکلیئر کریں  |
| 10:46 | class کے objects کس طرح بنائیں  |
| 10:48 | Ruby میں میتھڈس ڈفائن کرنے کے مختلف طریقے   |
| 10:52 | ایک مشق:  |
| 10:54 | ایک class Product ڈفائن کریں  |
| 10:56 | myvar کی ویلیوز حاصل کرنے اور myvar کے لئے ویلیوز مقرر کرنے کے لئے methods ڈفائن کریں             |

|       |   |
|-------|---|
| 11:01 | ویلیوز قائم کرنے کے لئے، "=" سائن استعمال کرتے ہوئے method ڈفائن کریں   |
| 11:05 | class کے آنجیکٹس کو وضاحت سے دکھانا اور مندرجہ بالا دو میتھڈس استعمال کرتے ہوئے ویلیوز سیٹ اور حاصل کرنا۔   |
| 11:12 | مندرجہ ذیل لنک پر دستیاب وڈو دیکھیں۔  |
| 11:14 | یہ اسپوکن ٹیوٹوریل پروجیکٹ کا خلاصہ کرتا ہے۔  |
| 11:18 | اچھی بینڈ وڈو تھ نہ ملنے پر آپ اسے ڈاؤن لوڈ کر کے دیکھ سکتے ہیں۔  |
| 11:22 | اسپوکن ٹیوٹوریل پروجیکٹ ٹیم:  |
| 11:24 | اسپوکن ٹیوٹوریلز کا استعمال کرتے ہوئے ورکشاپ چلاتی ہے،  |
| 11:27 | اور آن لائن ٹیسٹ پاس کرنے والوں کو ٹیوٹوریل دیتے ہیں۔   |
| 11:30 | مزید معلومات کے لئے، <a href="mailto:contact@spoken-tutorial.org">contact@spoken-tutorial.org</a> پر لکھیں۔   |
| 11:36 | اسپوکن ٹیوٹوریل پروجیکٹ، ٹاک ٹوائے ٹیچر پراجیکٹ کا حصہ ہے۔  |
| 11:39 | یہ بھارتی حکومت کے ایم ایچ آر ڈی کے آئی سی ٹی کے ذریعے قومی خواندگی مشن کی طرف سے حمایت شدہ ہے۔   |
| 11:46 | اس مشن پر مزید معلومات <a href="http://spoken-hyphen-tutorial-dot-org-slash-NMEICT-hyphen">spoken hyphen tutorial dot org slash NMEICT hyphen</a> Intro پر دستیاب ہے۔ |
| 11:56 | اس اسکرپٹ کا ترجمہ اور صدا بندی میں نے یعنی وجاحت احمد نے کی ہے۔ شامل ہونے کیلئے آپ کا شکریہ۔   |